ESD-TR-86-085

Semiannual Technical Summary

Distributed Sensor Networks

31 March 1986

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

Prepared for the Defense Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-85-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-85-C-0002 (ARPA Order 3345).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Thomas J. Alpert

Thomas J. Alpert, Major, USAF
Chief, ESD Lincoln Laboratory Project Office

Non-Lincoln Recipients

PLEASE DO NOT RETURN

Permission is given to destroy this document when it is no longer needed.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

DISTRIBUTED SENSOR NETWORKS

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

1 OCTOBER 1985 — 31 MARCH 1986

ISSUED 16 SEPTEMBER 1986

LEXINGTON MASSACHUSETTS

ABSTRACT

This report describes the work performed on the DARPA Distributed Sensor Networks Program at Lincoln Laboratory during the period 1 October 1985 through 31 March 1986.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

DISTRIBUTED SENSOR NETWORKS

I. INTRODUCTION AND SUMMARY

The Distributed Sensor Networks (DSN) program is aimed at developing and extending target surveillance and tracking technology in systems that employ multiple spatially distributed sensors and processing resources. Such a system would be made up of sensors, data bases, and processors distributed throughout an area and interconnected by an appropriate digital data communication system. The detection, tracking, and classification of low-flying aircraft has been selected to develop and evaluate DSN concepts in the light of a specific system problem. A DSN test bed has been developed and is being used to test and demonstrate DSN techniques and technology. The overall concept calls for a mix of sensor types. The initial test-bed sensors are small arrays of microphones at each node augmented by TV sensors at some nodes. This Semiannual Technical Summary (SATS) reports results for the period 1 October 1985 through 31 March 1986. Final live distributed tracking experiments and demonstrations with the test-bed elements deployed in the Lincoln Laboratory/Hanscom Field area are planned for the late summer.

Primary emphasis during this report period has been in the areas of: (a) site data collection, integration, and display; (b) cooperative distributed tracking with both acoustic and TV sensors; (c) test-bed communications; and (d) test-bed upgrades to improve reliability. The major accomplishments are summarized below. Sections II to V provide more extended summaries and additional technical details.

Accomplishments in the area of multisite integration include the design and testing of algorithms and a query and display capability for the DSN test bed. The algorithms use the same techniques for combining tracks from multiple sites as are used by the basic DSN tracking system. They have been implemented and tested in the form of multiple cooperating processes on a single computer as a step toward eventual implementation in test-bed nodes. They are designed to collect and integrate data from any set of network nodes using a data-collection tree that is user specified. The query and display system provides a color display with menus and interaction using a mouse. Display options include the network configuration, target tracks, the data-collection tree, and which nodes have contributed data to which tracks. The user can move and zoom the situation display.

Work in the area of cooperative tracking with multiple sensor types has emphasized the development of TV subsystems for the test bed and their integration with the tracker. TV pointing algorithms were refined. These algorithms accept lists of target tracks from the tracker, select a target, and generate pointing commands for the TV camera, which is on a remotely controlled mount. Techniques were developed to perform real-time TV and acoustic tracking experiments with simulated data. These techniques were used to test the integration of the TV system with the rest of the DSN test-bed system and to perform initial tracking experiments with multiple sensor types.

Up to the present time all test-bed experiments have utilized an Ethernet for internodal communications. DSN experiments at Hanscom Field and future Air Vehicle Survivability Evaluation (AVSE) field experiments will require nodes to be separated by several kilometers; beyond the range of an Ethernet. Therefore, with additional funding provided by the AVSE project, we have procured a commercial microwave radio communication system that will interconnect up to four nodes and have begun to integrate it into the DSN test bed. When this is completed the test bed will operate using a combination of microwave and Ethernet systems to interconnect the nodes. In addition, we completed the implementation and testing of a simple broadcast protocol for experimental Communication Network Technology packet radios that have been developed for DARPA by another Group at Lincoln Laboratory. Technology incorporated in those radios could be the basis for future DSN communication systems.

Reliability upgrades to the test bed included changes in front ends, improved tape recorders, and ruggedization of computers as summarized in Section V.

II. MULTISITE INTEGRATION

Each node in a DSN system maintains tracks for targets within range of its own sensors. Adjacent nodes may contain tracks for the same target. Some of these redundant tracks may differ from each other in detail and the entire set of tracks for a specific target will move from node to node within the network as the target moves. Redundant tracks must be merged together and users should be provided with nonredundant "full-network" tracks. This requires collecting and merging information from multiple sites and providing the integrated result to the user in a useful form.

Multisite track integration algorithms and a user query and display workstation for the DSN test bed were designed, implemented and tested during this report period. The multisite integration algorithm was implemented, and tested in the form of multiple communicating processes on a single UNIX system as a major step toward implementing it in the test-bed nodal computers. Testing was performed with prerecorded helicopter data and with simulated data for one and two helicopters and up to eight nodes. The query and display functions were implemented on a Silicon Graphics, Inc. (SGI) UNIX workstation with a color display.

A. MULTISITE INTEGRATION ALGORITHMS

Multisite data integration makes use of an acyclic data integration routing tree, rooted at a user display station and extending into the network to collect and integrate data from all nodes of interest. Tracks are sent inward from the leaves to the user with nodes combining redundant tracks as redundancies are detected. Standard track identifiers are used to determine when tracks are redundant. Track files presented to the user have had all redundancies removed and contain tracks from all the nodes in the data-collection network; not just a single node.

The structure and contents of the internodal messages used for data integration are similar to those of the normal target tracking messages, with extensions made to handle extra multisite integration information. The normal tracking messages contain local and common target state (position and velocity) vectors and error covariance matrices for one or more targets. The integration messages also include explicit lists of the nodes that contributed to each reported target at each time. In addition, provision is made to report nodes which fail to provide integration messages. As illustrated in the sequel, the extra information can be very useful in understanding system behavior.

The tracking nodes in the test bed operate with update periods of 2 s. The multisite integration algorithms operate with an update period which is an integer multiple of the tracking node update cycle. Two versions of the update process have been considered. In one case the integration processing cycle is started during the same tracking node cycle at all network nodes. In that case the multisite integration process in each node acquires tracks from the tracking process in the node at the start of the integration cycle. Multisite data collection and track combining is then initiated at the leaf nodes. Multisite integration messages propagate inward from the leaf to the user. The integration messages provided to the user contain the aircraft traffic status at the start of the cycle. In the other case the integration processing cycle is started during the same tracking node cycle at the leaf nodes of the data integration tree. As information flows inward toward the user the most up-to-date tracks at each node are used to update the integration message. The integration message provided to the user contains an estimate of the aircraft traffic status at the time the message arrives, although it may be partially based upon earlier node information that has been extrapolated to the present time. It is the first of these two cases that has been implemented and tested.

The multisite integration algorithms use the track combining algorithms that were previously developed for tracking. The multisite integration algorithm at each node uses the tracking algorithm to combine tracks found in multisite integration messages from other nodes with its own local tracks. Tracks that are otherwise unknown to the node are passed on without modification. The lists of nodes contributing and not contributing to the tracks and messages are also updated. The resulting algorithm has performed well thus far.

The integration algorithms have been implemented in the form of a main control process and a group of node processes that all execute on a single UNIX system. The node processes are interconnected by pipes corresponding to a multisite integration route tree. The main process reads parameter files, translates parameter files, and executes the node processes. The main process sends messages to the nodal processes to specify algorithm and route parameters. The algorithm parameters set uncertainty and threshold values used by the multisite integration algorithm. The route parameters specify the network route for multisite data collection and integration. Route information within a node process specifies only the nodes with which it interacts directly. It is a list of the nodes from which to expect messages and the one node to which it should send messages.

The multisite integration algorithms were tested using nodal track data previously recorded on floppy disks in the test-bed nodes during multinode tracking experiments. The data from the disks were transferred to UNIX files, one for each node in an experiment. The node processes read track information from these files and also receive multisite integration messages from other nodes through the pipes. Each node reads its input pipes, performs multisite integration, and writes to its output pipe. Multisite integration communication failures can be simulated by filtering the internodal pipes to remove some of the messages. The nodes can then detect missing messages by monitoring time stamps and can take the appropriate action, which includes no further waiting for old data and adding the missing message information to its output integration message. The last node in the integration route produces the multisite integration messages for the user. At the present time these user messages are saved in a file and later transferred to an SGI UNIX workstation where they are displayed by the user. Integration of the display workstation into the test bed will be completed during the next reporting period.

B. QUERY AND DISPLAY SYSTEM

A multisite query and display system has been implemented for the DSN test bed. It provides for a number of useful displays and options. These include showing present target locations and past histories, node locations, the data integration routing tree, network communication connectivity, and nodes contributing to specific target tracks. It also provides for zooming in or out on areas of interest. Tracks are depicted by error ellipses, based upon track covariance matrices provided in the multisite integration messages, that indicate the size and shape of estimated track uncertainties.

The display system requires several input files in addition to the multisite integration message stream. These include files that specify node locations, communication ranges, and sensor detection ranges. It also requires multisite integration route specification messages.

The display program is interactively controlled by means of menus and a three-button mouse. The buttons are used to invoke functions such as changing menus, selecting objects, recentering the screen, zooming in, and zooming out.

Figure II-1 shows an example of the situation display. Node locations are shown as small yellow dots. The data integration tree is indicated by the long blue isosceles triangles. Each triangle is an arrowhead pointing from a sending to a receiving node in the multisite integration route tree. Thirty-second track histories are shown for two targets flying from the left to the right. These are the two series of four small red error ellipses. In the case shown the integration cycle was five times the basic tracking cycle so that the time between ellipses is 10 s. The user can elect to display 30- or 120-s histories. The straight yellow lines radiating from the tracks to node locations indicate which nodes contribute to the track. The text at the right side is a help menu to remind the user how to use the mouse buttons. Pressing all three buttons toggles the display of this text. Other information on the display of Figure II-1 includes a labeled geographic grid and two alphanumeric strings that indicate the present time and the time of the last received track integration message. The small red arrow is a cursor used to pick options from other menus and select targets or nodes. Its position is controlled by the mouse.

Figure II-2 shows a display similar to that of Figure II-1 except that the display has been zoomed in and is for a later time during the same experiment. Note that the node in the upper right corner is not contributing to the rightmost track although it is close enough to be within detection range. Although it is not shown on this display, the user could also request to see the detection range of the sensor, shown as a circle centered upon the node, to confirm that the target is indeed within normal detection range.

The display in Figure II-2 can be used to infer that there is a malfunction in the tracking system at the upper right node, although the node is performing its multisite data collection and integration functions. This requires knowing that, normally, as illustrated in Figures II-1 and -2, the route triangles are solid, but when messages are not being received the route specification triangles are shown only in outline as in Figure II-3.

75932-1

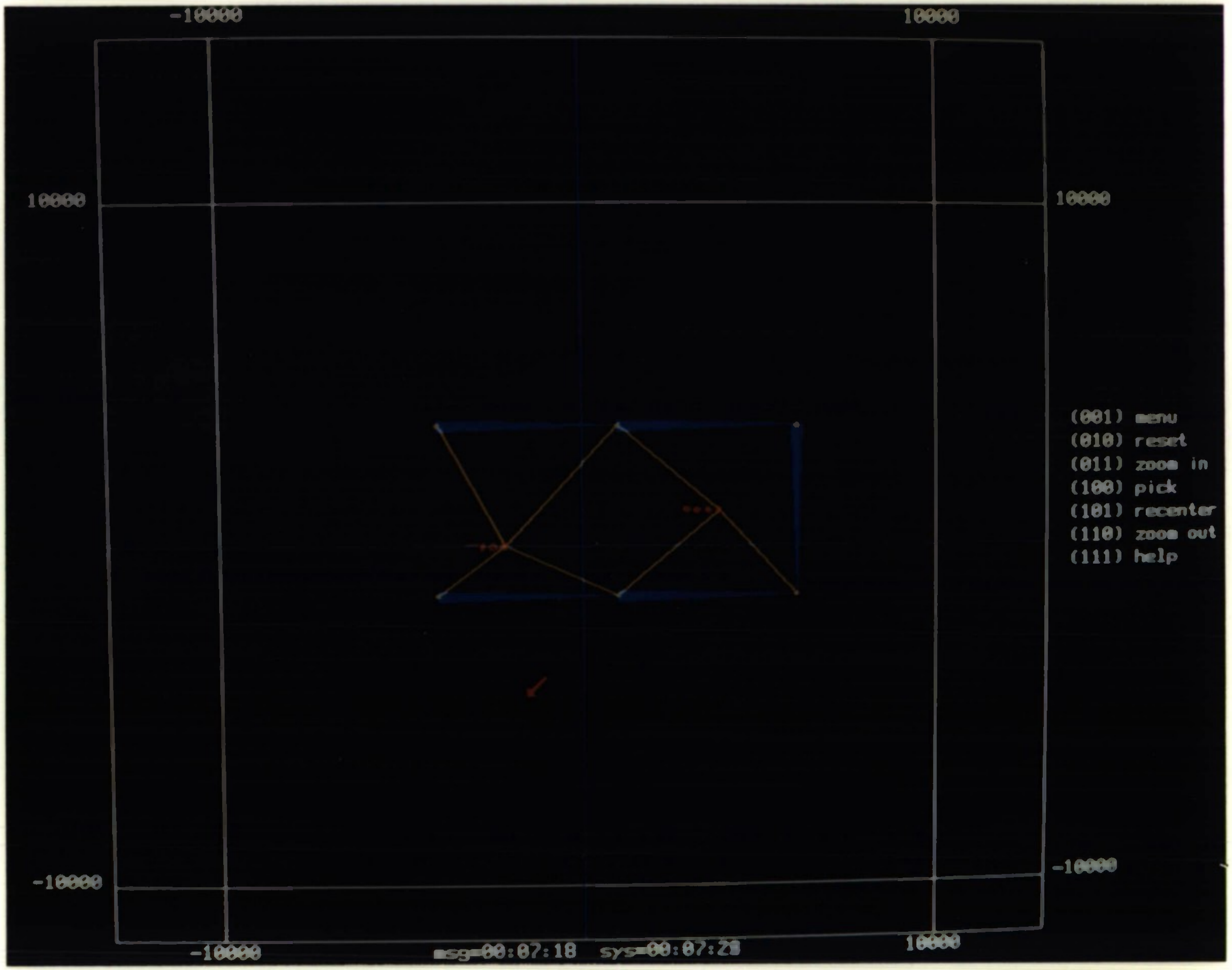


Figure II-1. Situation display showing six nodes, two tracks, data-collection route, and nodes contributing to the tracks.

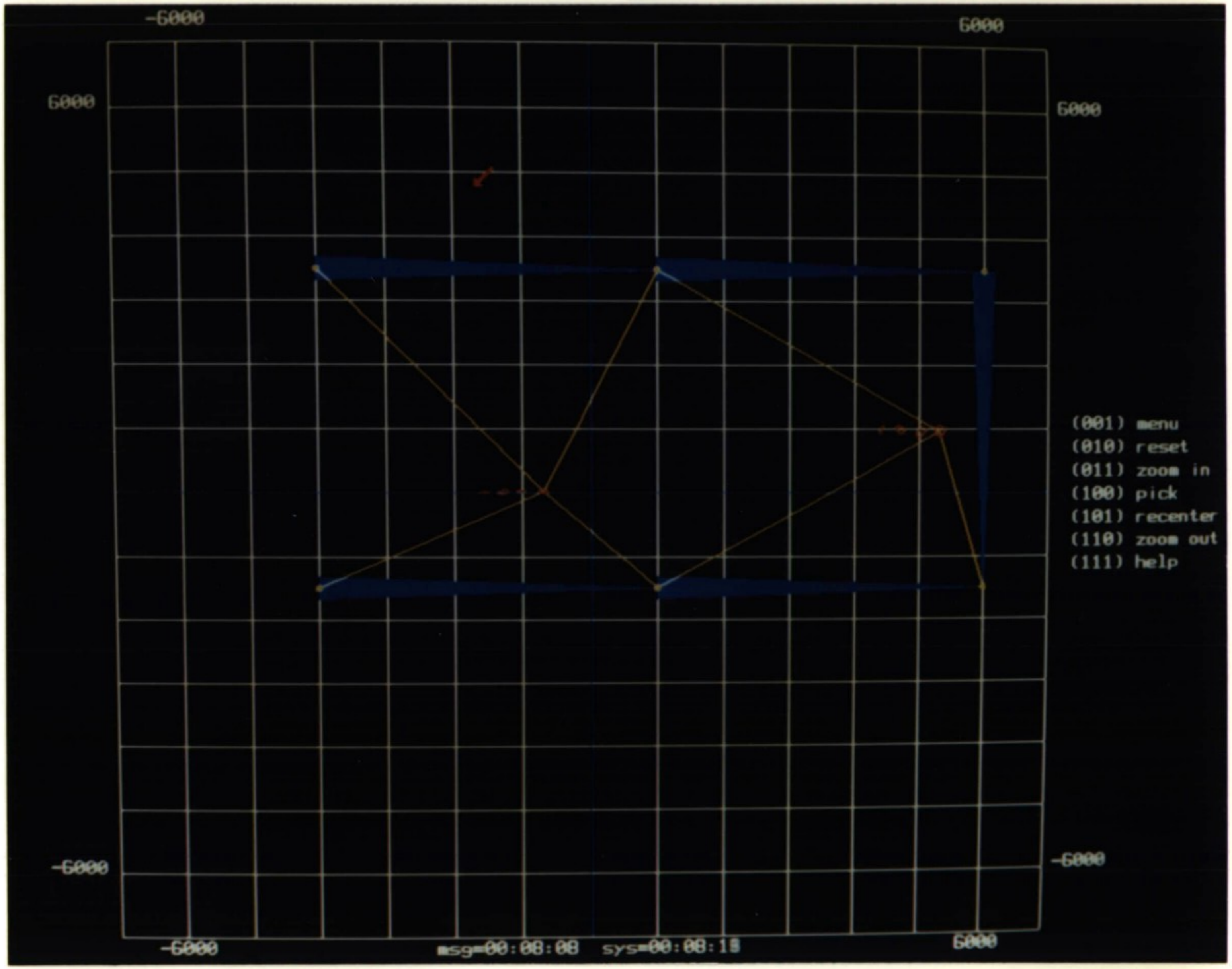


Figure II-2. Situation display similar to Figure II-1 but for later time and zoomed in for more detail.

75932-3

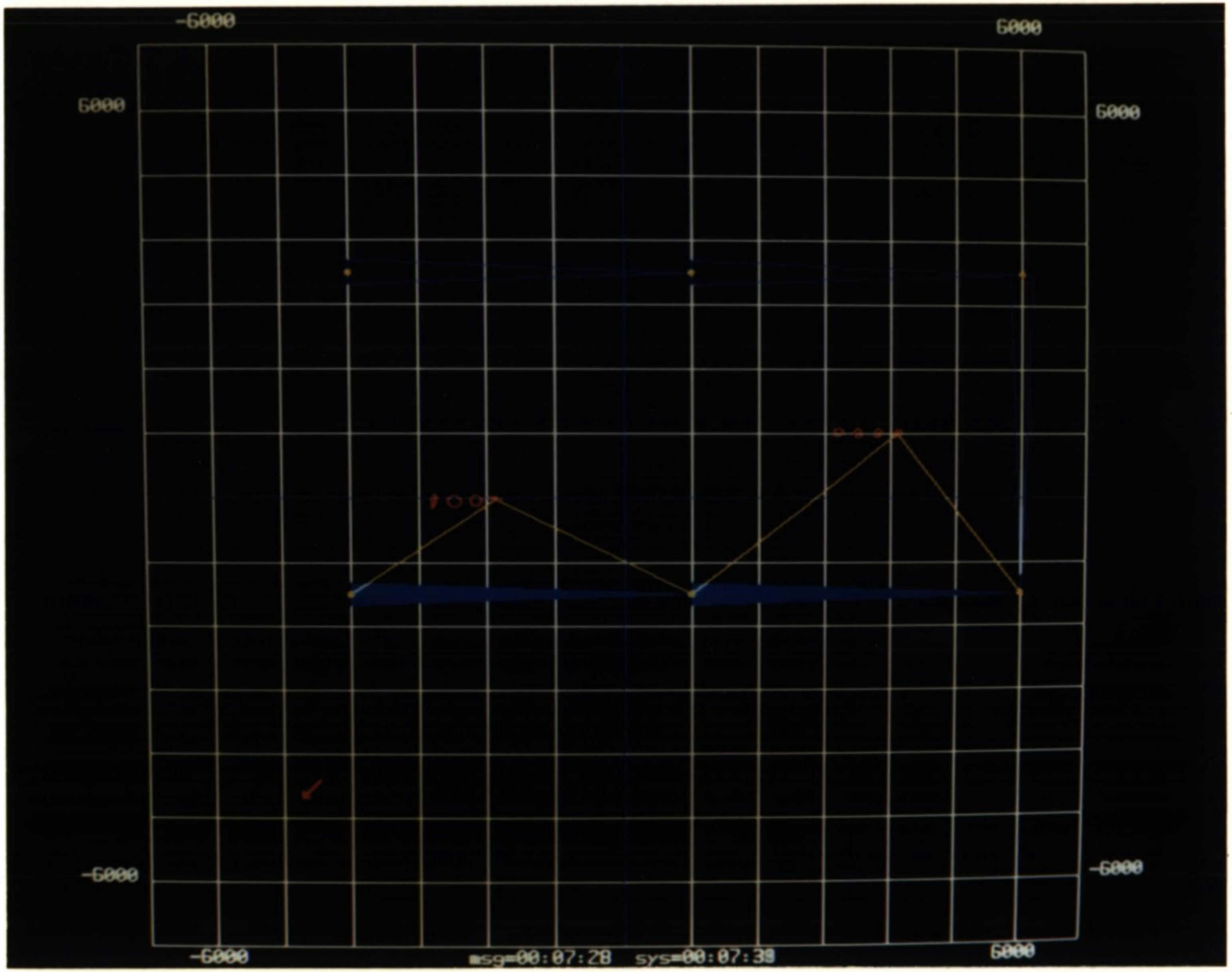


Figure II-3. Situation display similar to Figure II-1 but showing communication failure.

The situation in Figure II-3 is clearly more catastrophic than in Figure II-2. The user, located at the lower right-hand node, is receiving no multisite integration messages from the node at the upper right. Because of this no information can be collected from any of the upper three nodes. The user knows only that the multisite integration messages from the upper right-hand node are not being received. This is true even though two more of the yellow lines from tracks to nodes are missing. We cannot infer that the other two nodes have failed. The tracks from the lower nodes may contain information from the upper nodes but that cannot be known because the information was exchanged between nodes in the form of normal tracking messages and not multisite integration messages.

Returning to Figure II-2 we see that all communication links are working, that two of the nodes are contributing normally to the tracks, and one node is not. Thus we conclude that there is a problem with the tracking process in one of the nodes. This illustrates how a user interface can help with detecting and diagnosing system failures as well as providing surveillance information for users.

The multisite integration algorithms and display system have also been of value in detecting other forms of system failure such as tracking algorithm bugs. During one multisite integration experiment it was discovered that if three nodes initialize tracks simultaneously for the same target, it is possible that one node will assign a different track identifier than the other two. The result is two poor-quality tracks rather than a single good-quality track. The cause of this low probability of occurrence situation is now understood and the situation can be remedied.

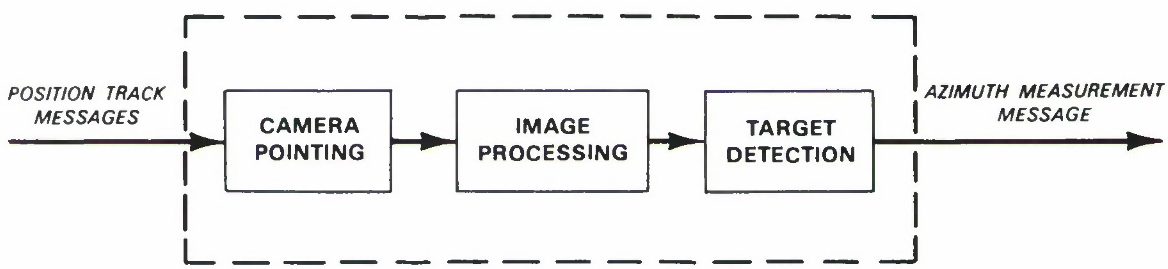
III. COOPERATIVE TV AND ACOUSTIC TRACKING

Work has continued on the development of test-bed capabilities to demonstrate the cooperative use of sensors with complementary properties. Acoustic surveillance and tracking capabilities have been developed previously and emphasis during this reporting period has been upon the development of TV subsystems and upon the integration of TV subsystems with the test-bed tracking system. The TV subsystems are being developed for use in the cueing mode. They will accept position cueing messages from a DSN tracking node, use the information in these messages to position a TV camera, and return video-derived azimuth measurements to the tracking node. The following summarizes progress made in the areas of: (a) algorithm development, (b) algorithm testing, (c) hardware, and (d) system software.

A. ALGORITHM DEVELOPMENT

The algorithm development effort has been concentrated in two areas: enhancements to the camera-pointing algorithm and development of a TV subsystem simulation.

The TV node algorithms perform three operations as illustrated in Figure III-1. The first of these is camera-pointing which takes as inputs: (a) the estimated position and velocity of N targets and (b) the current camera azimuth. Based on this information, the algorithm selects a target and a camera slew procedure (policy) for pointing the camera toward the target and generates appropriate camera control commands. The target and camera policy selections are based upon four factors. These are: Time-to-Catch (how long will it take the camera to slew to the target), In-Track-Time (how long can the camera follow the target), Acquisition-Range (camera-target range at acquisition), and Acquisition-Azimuth (target azimuth at acquisition).



75932-4

Figure III-1. TV subsystem algorithm elements.

The camera-pointing algorithm was improved in two ways during this report period. First, a comprehensive camera policy generator was developed. As illustrated in Figure III-2, it computes a list of feasible camera-pointing policies for every target. This list, which typically includes only 2 to 5 policies, is generated by pruning a more exhaustive list of 34

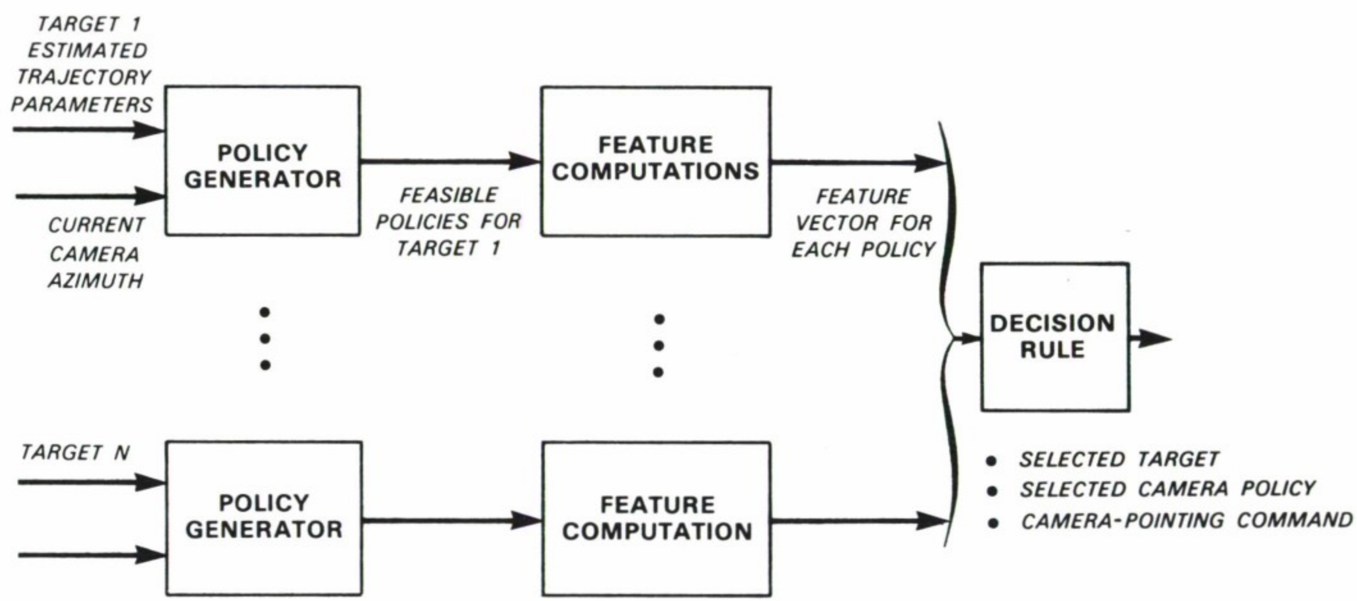


Figure III-2. Complete camera-pointing algorithm.

75932-5

possible camera-pointing policies on the basis of simple rules involving target motion and camera position. Features used for final target and policy selection are computed only for the few feasible policies that survive the initial pruning.

The large number of possible camera policies arise because of three possible target/camera conditions that can occur. First, the target may fly approximately over the camera, thereby moving above the camera field of view. Second, the target may fly through a region where the camera cannot slew fast enough to keep up (an untrackable region). Finally, the camera dead-zone may prevent the camera from slewing to a desired azimuth in one direction. The significance of these three factors is illustrated in the example depicted in Figure III-3.

In Figure III-3 the camera is initially pointing West and the target is traveling South on a track that will bring it to within d = 1 km from the camera. The camera cannot slew through its dead-zone (which begins/ends at 105°/115° in azimuth). The camera cannot slew fast enough to keep track of the target when the target is inside the untrackable region which is also indicated in Figure III-3.

Two feasible policies and their associated features are shown in the figure. For Policy 1 the camera slews clockwise and acquires the target quickly but is able to maintain track for only 6 s, at which time the target enters the untrackable region. For Policy 2 the camera slews counterclockwise (CCW) and acquires the target only as the target emerges from the camera dead-zone. The in-track-time for Policy 2, however, is superior to that of Policy 1. The current camera-pointing algorithm selects Policy 2 in this case on the basis of the longer time-in-track that it will provide.

Other camera policies become feasible as the distance d is varied. If the distance d is made smaller, the untrackable region becomes larger and overlaps the dead-zone resulting in other feasible camera policies (e.g., slew CCW and wait at the end of the untrackable region). If the distance d is made approximately zero, a "target-over-camera condition" is declared resulting in still other feasible policies (e.g., acquire inbound/outbound).

The computations required by the camera-pointing procedure described above take approximately 0.8 s per target to execute in the TV subsystem. We have also implemented a modified version of the algorithm that reduces this time to 0.2 s during most of the TV subsystem run time.

The elements of the modified algorithm are shown in Figure III-4. The first time a position message arrives with the tracks of N targets (i.e., when the TV node is first enabled), the full camera-pointing algorithm is executed resulting in the selection of a target and a camera-pointing policy, and in the issuing of a camera-pointing command. In addition, the identifier (an ID provided by the cueing DSN tracking node) of the selected target and its associated in-track-time are saved. Thereafter, as new position messages arrive, the full camera-pointing algorithm is executed only if: (a) the track corresponding to the previously selected target has been dropped by the cueing DSN node, (b) the in-track-time

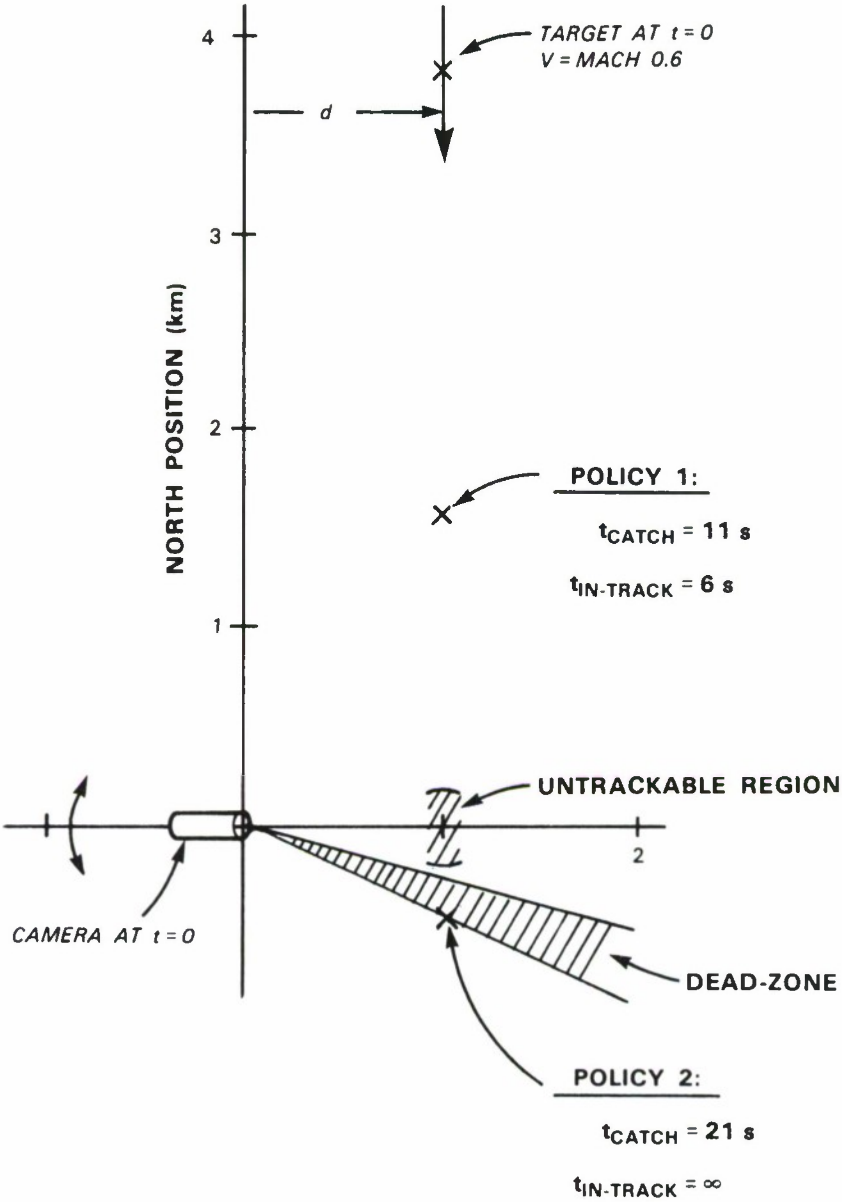


Figure III-3. Camera-pointing example.

75932-7

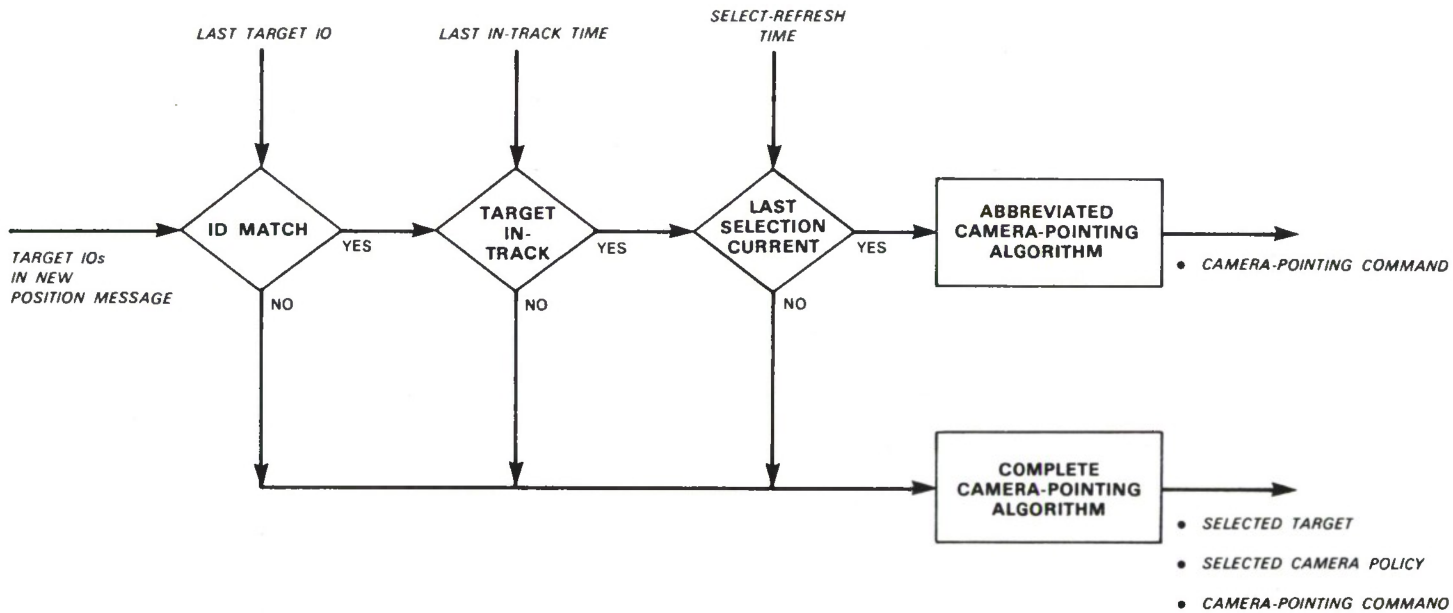


Figure III-4. Selection of abbreviated/complete pointing algorithm modes.

period of the previously selected target has expired, or (c) the previous target selection is declared to be obsolete on the basis of the "select-refresh-time" parameter (currently 15 s). The select-refresh-time parameter is used to force the system to periodically reconsider its selections, even if other criteria do not require it to do so. In all other situations an abbreviated pointing algorithm is executed. The abbreviated algorithm uses the previously selected target and camera-pointing policy. Target and camera-pointing policy selection functions are bypassed. Given the target ID and camera-pointing policy, the abbreviated algorithm generates camera-pointing commands using the most recent target state estimate provided by the tracker.

The other algorithm task addressed during this reporting period was the development of a technique for real-time combined acoustic and TV experimentation with simulated data. This capability was required for system integration tests and algorithm tuning since it provides for controlled and easily repeatable experimental situations. As illustrated in Figure III-1, the TV node performs three sequential operations: camera pointing, image processing, and target detection. The simulation technique which has been developed is based upon replacing the last two operations (image processing and target detection) with simulation algorithms. The resulting TV subsystem simulation module, illustrated in Figure III-5, can be used to adjust TV parameters (e.g., elevation and zoom) before an experiment and to predict the expected performance. All other components of the TV subsystem are exactly the same as for live real-time experiments. The tracking nodes use simulated acoustic data; a previously developed capability. With that exception, the tracking nodes are also exactly the same as for live real-time experiments.

The inputs and outputs of the TV simulation module shown in Figure III-5 are identical in form to those for live experiments. The inputs are position track messages received from a tracking node. These are fed into the camera-pointing algorithm which in turn drives the TV camera. The outputs are: (a) azimuth measurements of the simulated target which go to the tracking node and (b) a display of the simulated target which is superimposed on a TV monitor display.

Figure III-5 shows major elements of the TV simulation algorithm. First, a target dynamics simulator computes the true target position and orientation in three-dimensional space. The aircraft is assumed to have zero pitch, yaw and roll, and to move along a straight path at constant velocity. This is a special case of the paths that can be used to generate simulated acoustic data.

Second, a plane-of-view (POV) simulator calculates target characteristics as they would appear in the plane of a TV frame. These characteristics include: the azimuth and elevation of the target, the location of the center of the target in the POV, and the projection of the target into the POV.

Third, a measurement generator produces a noisy azimuth measurement when two conditions are met: (a) the azimuth and elevation of the target and of the TV camera are

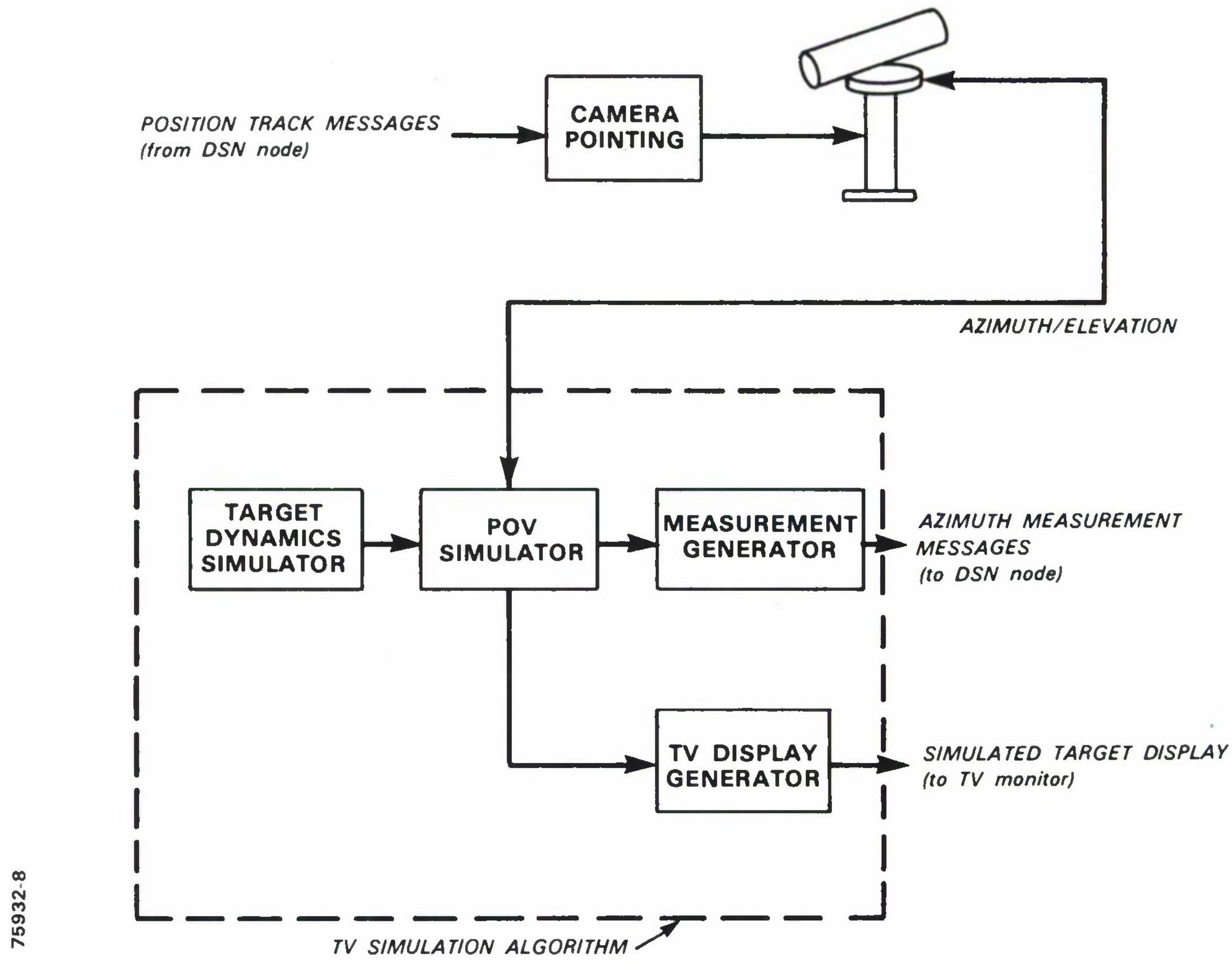


Figure III-5. TV simulation module for real-time experimentation.

such that the target is within the field-of-view of the camera and (b) the projection of the target into the POV is such that at least 7 pixels are covered (i.e., the target is sufficiently large to trigger a measurement). If these two conditions are met, an azimuth message is prepared and sent via the Ethernet to the tracking node that is providing cues to the TV subsystem.

Fourth, a TV display generator provides commands to the image processor to display the aircraft projection and the simulated azimuth measurement upon a live video display. The target and measurement displays appear only when they are within the field-of-view of the camera. The measurement appears only when the target projection is large enough to trigger a measurement.

B. ALGORITHM TESTING

The TV Subsystem Simulation was used to simulate the scenario shown in Figure III-6. In this experiment, two nodes with acoustic sensors are located 5 km apart and the TV node is deployed at the midpoint of the baseline connecting the acoustic nodes. The target is assumed to fly North at Mach 0.1 at a constant altitude of 350 m. The camera is initially pointing due South at a 10° elevation with the zoom set to produce a 15° field-of-view. With this situation the TV subsystem chooses to measure the target as long as possible as it is coming toward the TV. Once it rises above the field-of-view of the camera, the camera is slewed 180° to wait for it to reenter its field-of-view from above. Figure III-7 shows a comparison of the position tracks obtained with and without TV azimuth measurements.

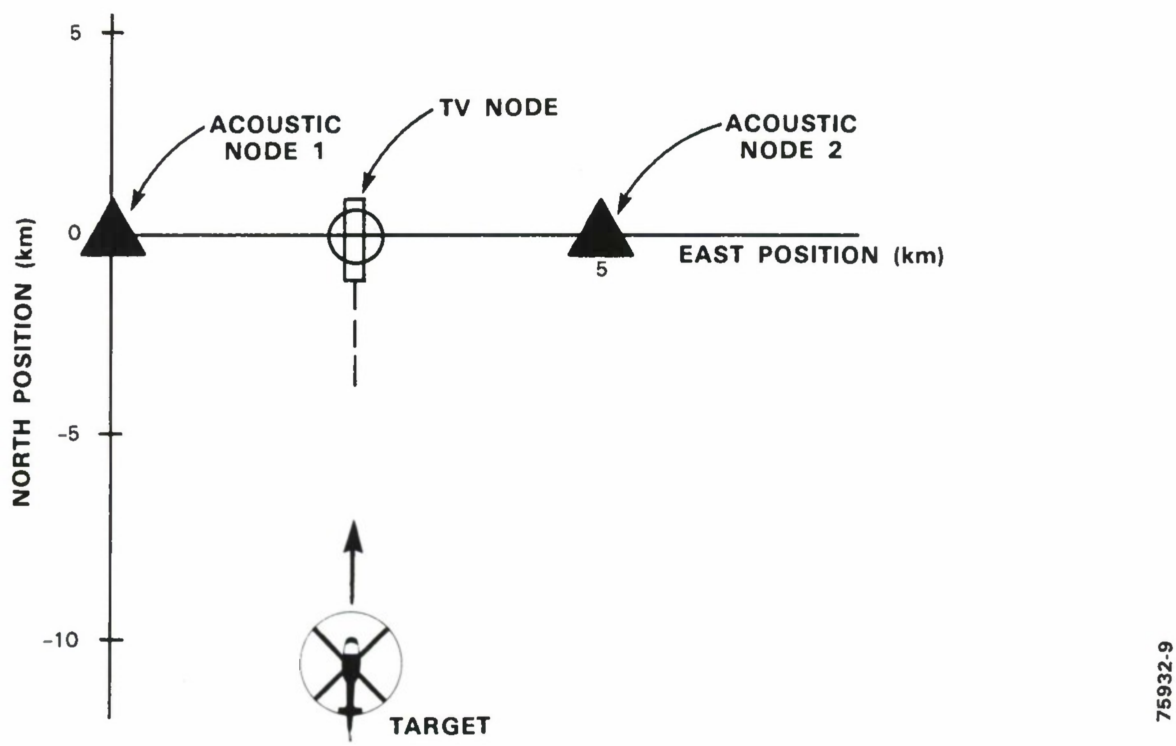


Figure III-6. Simulated TV/acoustic tracking scenario.

Without TV measurements [Figure III-7(a)], large error ellipses are obtained at the beginning of the track where the circular-error-probable is several hundred meters. Furthermore, throughout the track, the error ellipses remain large in the East-West direction as a result of the sensor/target geometry.

With TV measurements [Figure III-7(b)], the first two track points (Points 1 and 2) still show large errors because, initially, only acoustic measurements are available for tracking.

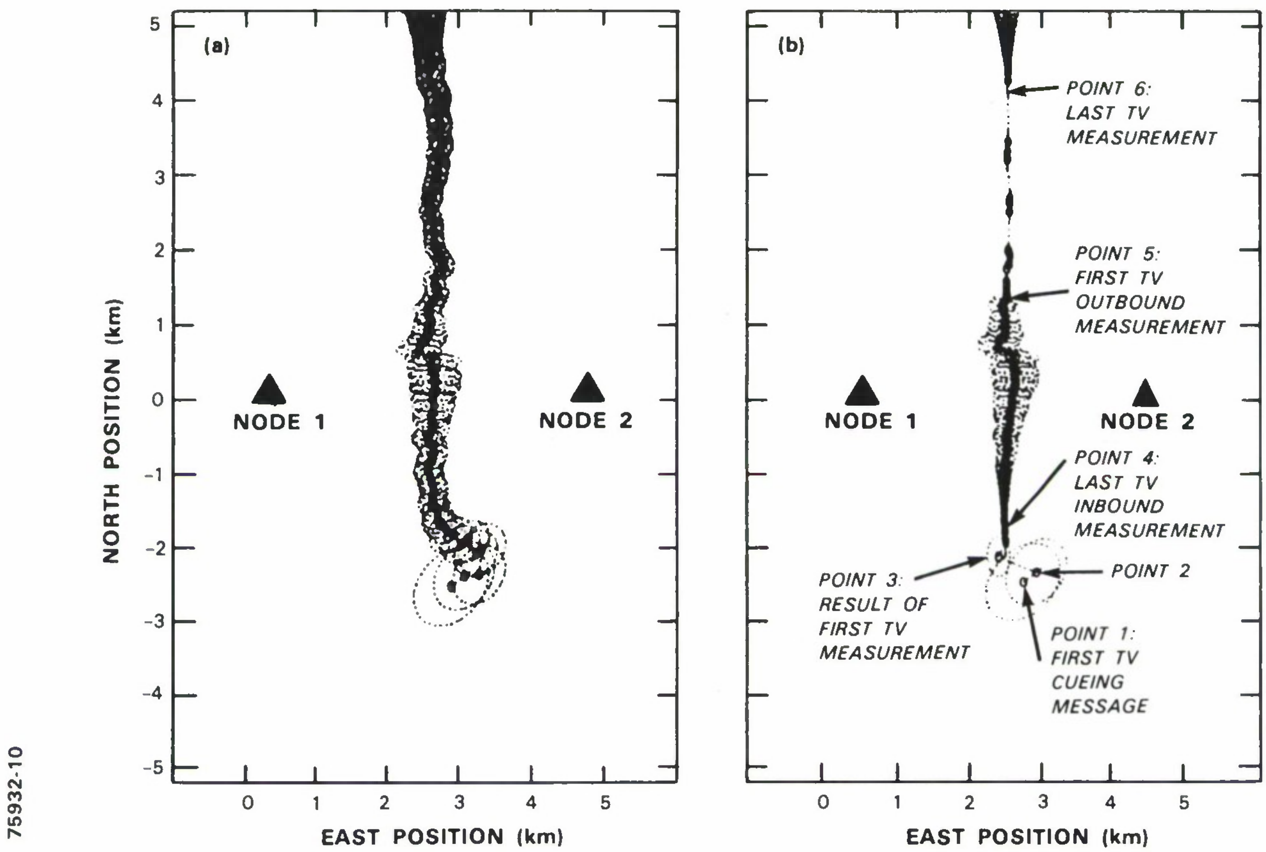


Figure III-7. Position track and estimated error ellipse comparisons: (a) without TV measurements and (b) with TV measurements.

However, when the first TV measurement is processed (Point 3), the error ellipses show a significant reduction. The reduction in the size of the error ellipses continues after the last TV measurement is recorded (Point 4) since the Kalman filter can make use of past track accuracy.

As the camera slews and waits to acquire the target on its outbound path, error ellipses grow to the levels obtained with acoustic-only tracking. When the outbound target enters the top of the field-of-view of the camera (Point 5), error ellipses decrease again and remain small until after the last TV measurement is obtained (Point 6).

Figure III-7 shows the error ellipses corresponding to the error covariance matrices generated by the extended Kalman filter in the tracker. Figure III-8 shows actual errors with and without TV measurements. The actual errors could be calculated since the experimenter knows the aircraft track that was used for data simulation. The figure shows the length of the vector from the true position to the position estimated by the tracker. With TV measurements, the error drops dramatically at the beginning of the track and the rate

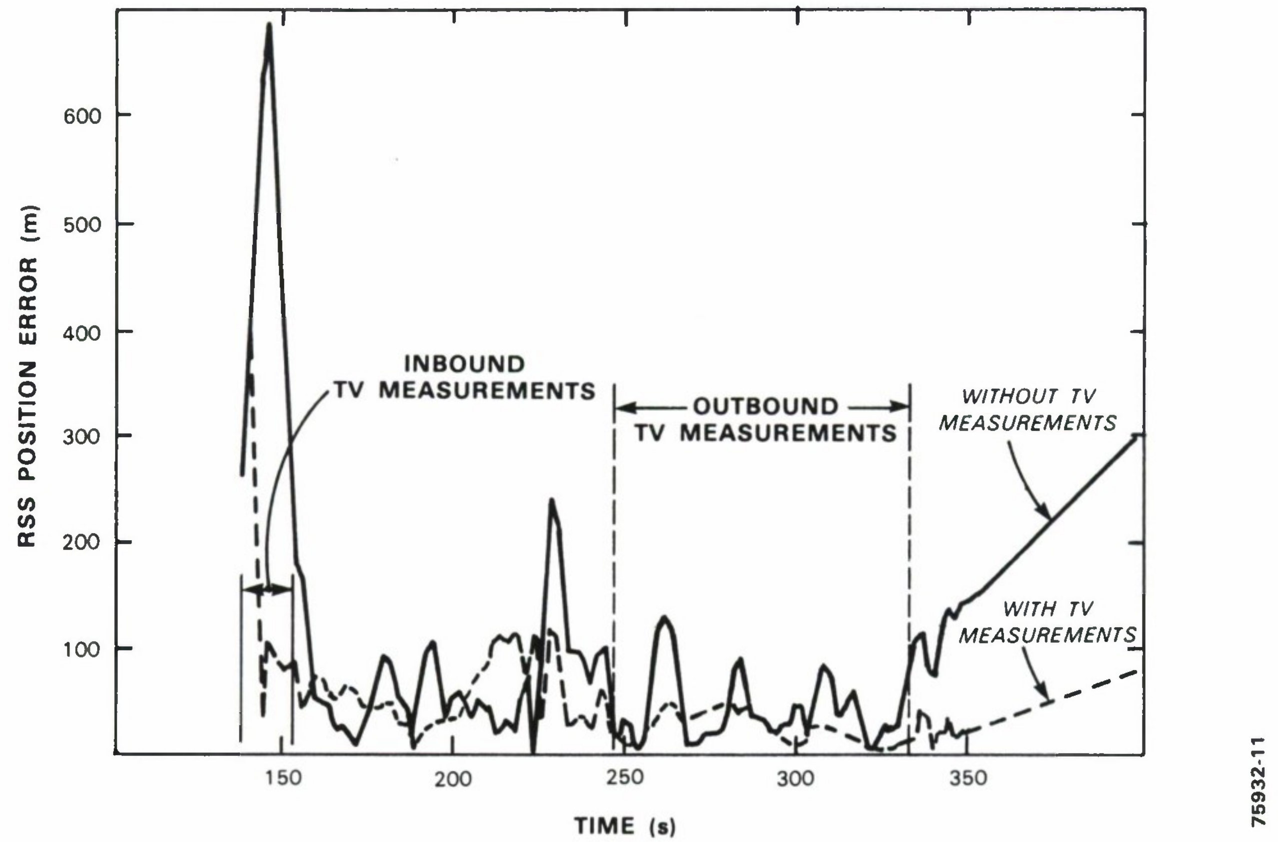


Figure III-8. Actual position errors with and without TV measurements.

of error growth at the end is significantly reduced. The simulated acoustic azimuth measurements used to obtain these results included rms measurement errors of 1°. Somewhat larger acoustic errors are expected in practice and the relative improvement due to TV measurements should therefore be larger. Future experiments will include tests with larger acoustic measurement errors to confirm this expectation.

This experiment represents a system integration milestone and is a first system tuning step toward demonstrations of how to use distributed sensors with complementary characteristics to achieve improvements in overall system performance. The field-of-view limitations of the TV sensor are complemented by the omnidirectional acoustic sensors used to initiate tracks, provide cues to the TV subsystem, and maintain tracks while the TV sensor slews and the target is passing overhead. The precision limitations of the acoustic sensors are supplemented by the TV sensor that provides high-accuracy azimuth measurements even at long distances. The result is more accurate and longer tracks.

C. HARDWARE

Hardware efforts have concentrated on the assembly of a second TV subsystem that will be installed in one of the test-bed vehicles. That TV subsystem will be remotely deployed and used in the field. The camera is mounted on a rugged but portable tripod and all control/signal cables to the camera-mount assembly are stored on and deployed from a single cable reel. The cables are presently 250 ft long but, if required, can be increased up to 1000 ft.

All the hardware for the new TV subsystem was acquired and assembled and preliminary tests were conducted in the laboratory. Operational shakedown tests and calibrations will be conducted during the next report period.

D. SYSTEM SOFTWARE

Substantial changes have been made to the software environment in which the TV application algorithms reside. These changes are illustrated in Figure III-9 which shows both the new and the previous TV software configurations. The new configuration enhances TV subsystem capabilities in three ways.

In the previous TV configuration, input commands and parameters were entered into the TV application algorithms only by typing commands and parameter values at a terminal. These commands and parameters were then sent by the User Interface Program via the Ethernet to the TV node. We have now added software in the TV node to allow the option to send command messages from the disk files on the host computer or workstation. The command sequence does not change very often so this capability has greatly reduced the amount of manual interaction that is required and has correspondingly reduced operator errors. It is now necessary only to carefully prepare and review the files to avoid operator errors.

The initial TV software configuration employed only one M68000 CPU to perform all TV node functions. Software and hardware changes have been made to implement a multiprocessing environment for the TV subsystems, resulting in greater storage and processing capability. This involved partitioning the TV algorithms into multiple processes and integrating them with NRTS (Node Run-Time System), the run-time system that is used in the other multiprocessor tracking nodes in the test bed. At the present time two M68000 CPUs are used in the TV node: P1 (Figure III-9) performs most system-related functions while P2 contains the applications code. We also have the option to add another CPU if that is required.

The third software change involved the frame buffer used to transfer image data to the M68000 CPU. The initial frame buffer has been replaced with a unit which offers the advantages of doubling the pixel data transfer rate and using memory-mapped I/O. Software changes in the host M68000 code were required to accommodate this increased capability.

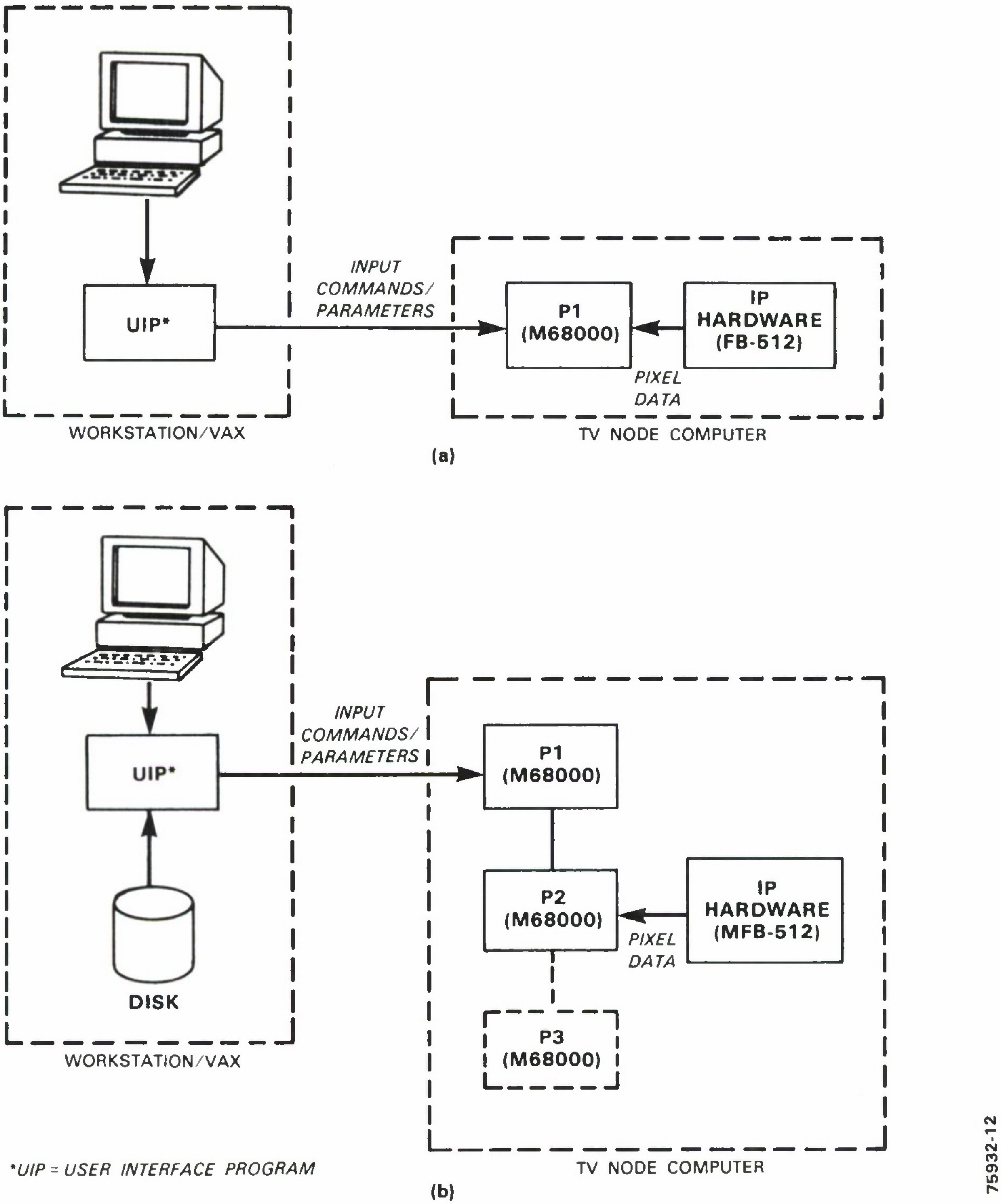


Figure III-9. TV subsystem software configurations: (a) single processor configuration and (b) multiple processor configuration.

IV. COMMUNICATION SYSTEMS

Progress has been made along two fronts in the area of DSN communications: (a) procurement and integration of commercial microwave radio equipment to support field experiments and (b) implementation of broadcast software for the experimental Communication Network Technology (CNT) radios developed by Group 86.

Live tracking experiments in the field are being planned as part of the transition of DSN technology to the Air Vehicle Survivability Evaluation (AVSE) program at Lincoln Laboratory. These experiments require a reliable field deployable communication system to interconnect three nodes plus a user interface workstation. Radio communication equipment for this purpose has been procured with funds provided by the AVSE project. This equipment is being integrated into the test bed and will also be used to support live DSN experiments near Lincoln Laboratory during this summer as well as subsequent AVSE field experiments. A software driver for the radio interface has been written and we are now proceeding to implement message forwarding software that will allow us to interconnect multiple Ethernet systems with this radio equipment and thereby support long baseline multinode experiments.

Other communication effort has centered upon the experimental CNT radios developed by Group 86 of the Lincoln Solid State Division. We have now completed the implementation of a simple broadcast protocol for the CNT radios. Those radios are experimental versions of advanced packet radios that might be considered for use in a future DSN system.

A. MICROWAVE RADIO SYSTEM

A transportable digital microwave system has been specified and procured to meet both DSN and AVSE internodal communication requirements for field experiments. The radio and multiplexer equipment is in hand. The delivery of the computer interfaces to the test-bed nodes (procured from a separate vendor) is expected by early May.

The basic radio requirement is to provide 50-kb/s bidirectional data links among four sites separated from each other by up to 25 km. In addition, the system must utilize standard military communication frequency bands and provide an easy way to select different operational frequencies so it can be adapted for use at different field sites. These requirements have been met with off-the-shelf hardware, and a substantial cost saving has been realized by means of an innovative variation on the standard configuration, as explained below.

The radio equipment is the Loral Terracom TCM-6 Portable Microwave System, featuring tripod-mounted antennas and electronics, together with Tau-Tron digital multiplexers that will be rack-mounted in the mobile equipment trucks. In normal operation a full-duplex radio is used at each end of each link, producing a bidirectional 1.544 Mb/s "T1 carrier" conforming to commercial telephone network standards. The T1 signal is formed by combining multiple lower-rate signals in commercial multiplexing hardware. A four-node ring network configured in this way would require eight antennas, eight transmitters, eight receivers, and eight multiplexers.

Figure IV-1 illustrates the plan for configuring the radio system to achieve substantial equipment savings. This network still requires eight (relatively low-cost) antennas, but the number of transmitters, receivers, and multiplexers is cut in half relative to the normal configuration. The digital T1 carrier from each multiplexer is divided into a transmit side and a receive side, with one corresponding radio transmitter and one receiver at each node, sending microwave signals clockwise around the ring. Two frequencies F_1 and F_2 are required, as in any full-duplex radio system, to avoid RF self-interference at each site. The radios operate in the military band extending from 7.125 to 8.4 GHz, and the selection of specific frequencies F_1 and F_2 must be coordinated at each field site. The multiplexers are of the "drop-and-insert" type, which forward the T1 carrier unchanged except for data removed or inserted on selected subchannels. For example, Site 1 in Figure IV-1 transmits directly to Site 2, while transmissions from Site 2 to Site 1 are forwarded through Sites 3 and 4.

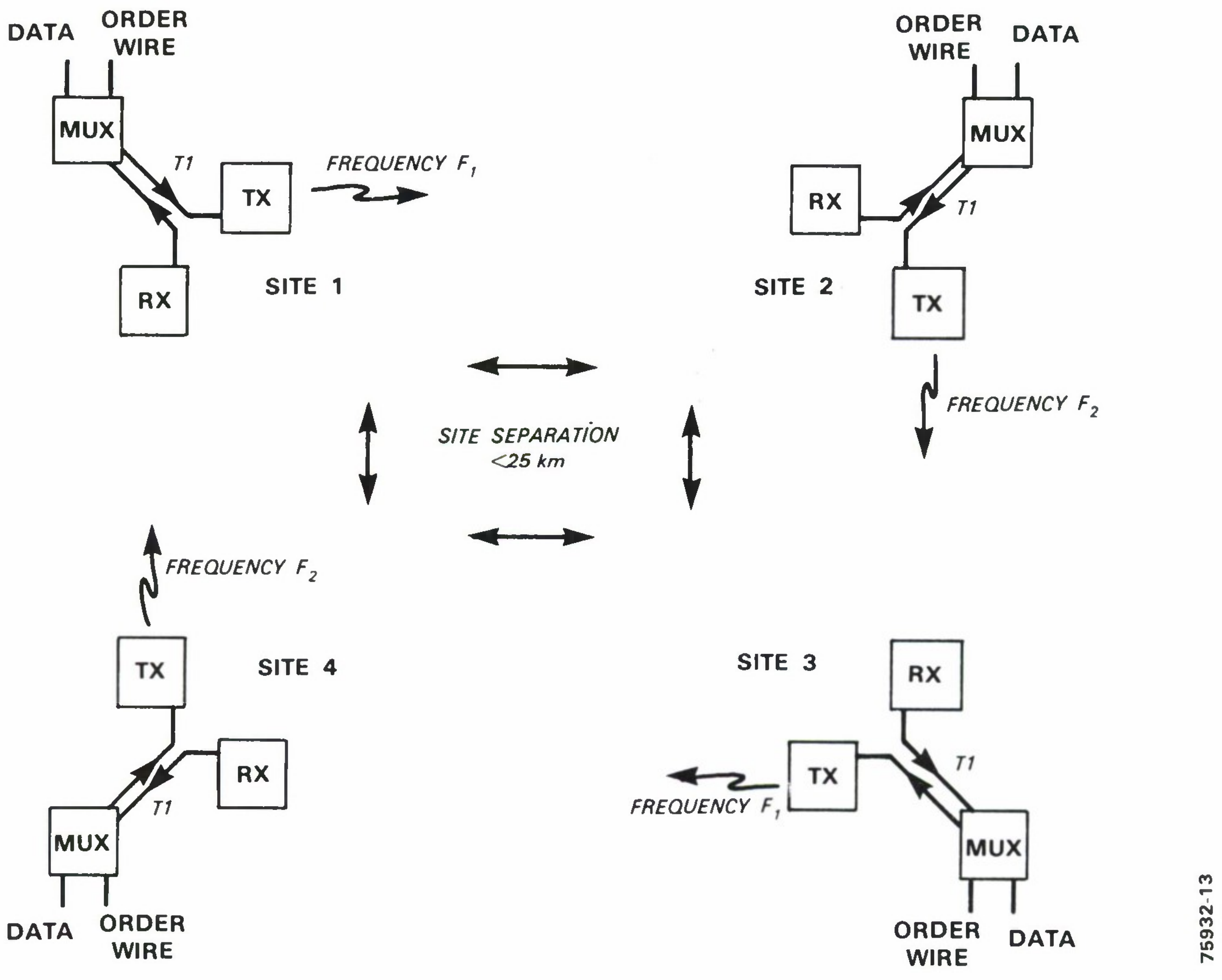


Figure IV-1 Portable microwave radio ring network.

To set up the radio communication system in the field, each leg of the four-node network in turn will be temporarily configured as a full-duplex link using a single antenna at each end. Antennas will be precisely aligned and error-free digital transmission will be checked out under these conditions. The network will then be reconfigured as shown in Figure IV-1. As an aid in both network alignment and fault isolation, two digital test sets and two RF power meters have been procured.

The subchannel digital interfaces provided by the communication system multiplexers are RS-449 synchronous 56 kb/s plug-in units. Digital interface boards for the test-bed Standard Nodal Computers (SNC) are being separately procured from SBE, Inc. of Concord, California, and are scheduled for delivery in May. The SNC interface boards were specified to minimize the software development effort required to integrate the microwave communication system into the test bed.

Acceptance tests have been successfully run on a prototype SNC interface board. The tests exercised all board functions and the transfer rates were higher than the specifications required. The interface between the board and the nodal computer met specifications, which are intended to minimize interrupts. The board is message oriented and interrupts the nodal operating system in only three situations. There is an initial interrupt to signal that the board is initialized. Thereafter an interrupt is received whenever a new message is received or when a requested message transmission is completed.

The test bed will use four channels on the SNC interface board, one channel for each DSN node equipped with a microwave radio. Each node will transmit on one channel and receive messages from the other three radio-equipped nodes on the other three channels. The interface provides substantial on-board buffering in the form of ten 1024-word buffers for each direction for each of the four channels.

Communication between the nodes and the interface board is by means of a four-item structure for each channel. The items are: a channel status byte, a transmit or receive flag, a byte count, and a pointer to a data buffer. To send or receive a message, the operating system (NRTS) sets the direction flag, a buffer pointer, a byte count if transmitting, and finally clears the status byte to start the transfer.

The interface board continuously polls status bytes. When a zero value is noted and the flag is TRANSMIT, the interface board initiates the data transfer from the user buffer into one of its on-board buffers if there is an on-board buffer available for that channel. If no buffer is available, the request will be honored only after a buffer becomes free. As soon as the data have been moved, the status byte for the channel is set to reflect that the transfer is completed and status bits are set indicating errors if any. An interrupt to NRTS is then generated. When a zero value is noted and the flag is RECEIVE, a message will be copied into the user buffer as soon as a message is available. The number of bytes in the message is stored in the count field, the status bits are set and an interrupt is generated.

A software driver and NRTS changes to handle the interface boards are designed and partially written.

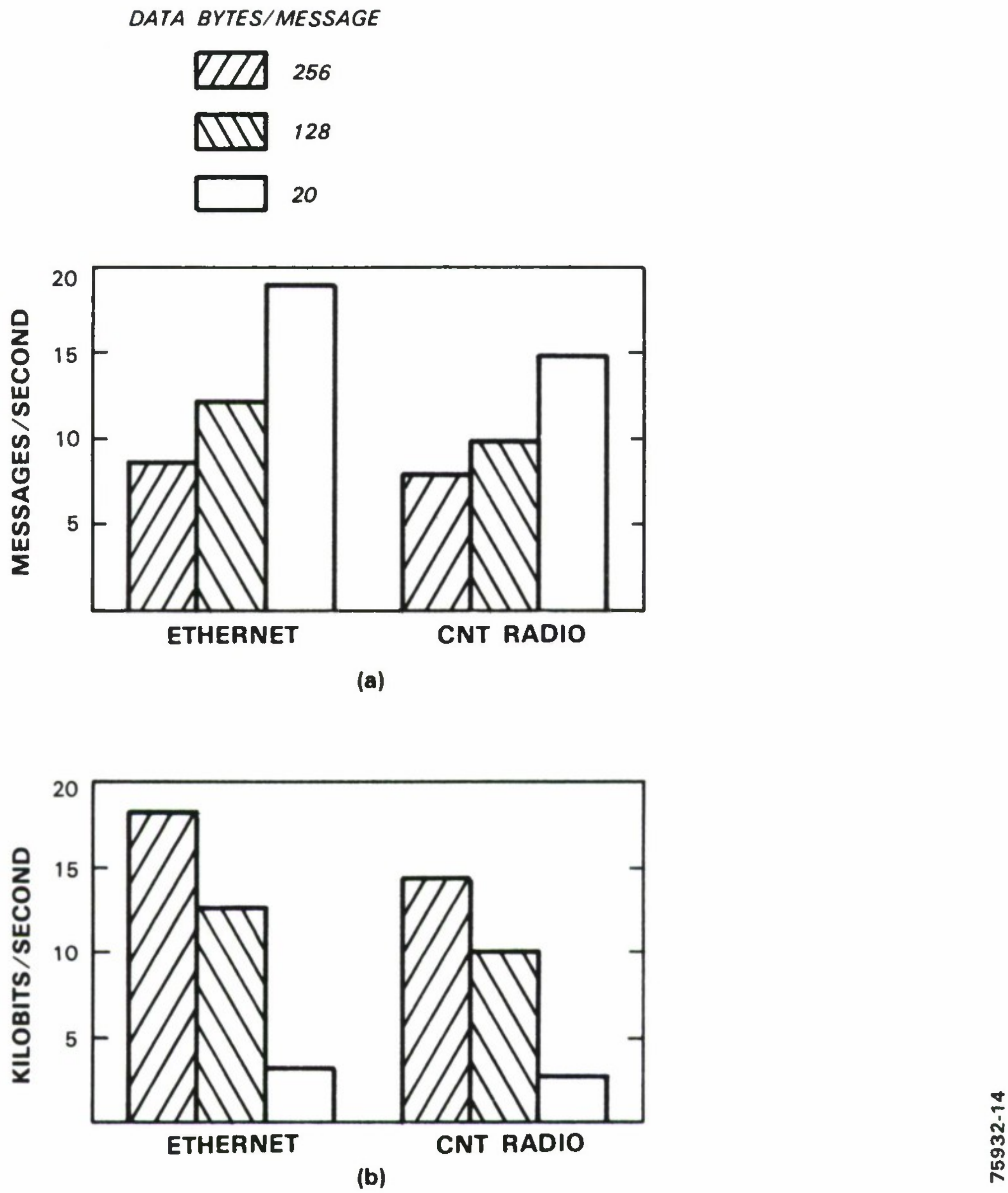


Figure IV-2. Internodal message throughput rates: (a) messages per second and (b) kilobits per second.

B. COMMUNICATION NETWORK TECHNOLOGY RADIOS

The implementation of a message broadcast protocol has been completed for the Communication Network Technology (CNT) radios developed by Group 86. Broadcast times are random, with the broadcast aborted when a reception is in progress at a scheduled broadcast time. Since DSN applications can tolerate some lost tracking messages, the broadcast protocol does not include positive message acknowledgment. This reduces communication overhead since each broadcast message is intended for several sites, each of which would be required to transmit acknowledgments.

Two application level programs were developed and used to test the implementation. The test setup in both cases consists of two DSN nodal processors interconnected by two radios. The nodal processors control and interact with the radios through Radio Unit Interface (RUI) boards that have been developed for that purpose.

One of the test programs provides an interactive message-oriented link between the two nodal processors. A line typed on the console of one computer is broadcast to the other where it is printed on a CRT display. The link operates equally well in either direction.

The other program is a traffic generator and statistics gathering package to provide more stressful testing. This program has provisions for sending messages of three different sizes: 20 bytes, 128 bytes or 256 bytes. Any number and mix of messages may be sent between the two test computers. Each computer can send a different number and mix of messages. Traffic can be in one direction only or in both directions. At the end of each test the user is provided with the elapsed time and the number of messages of each length received without error. Figure IV-2 shows the results of some tests using the traffic generator.

Two series of tests were run. For one, the traffic generator used the radios for internodal communication. In the other, an Ethernet was used as the actual communication link between the nodes. The same standard node operating system, NRTS, and message passing mechanisms were used in both cases. The only software difference was the use of the Ethernet driver in one case and the RUI driver in the other.

Several things are clear from the figure. First, throughput, in terms of either messages or bit rate, is comparable in the two cases although the RF signaling rate for the radio (~90 kb/s) is more than an order of magnitude less than that for the Ethernet. The slight decrease in performance for the radio is due primarily to the extra complexity of the driver required. Second, the bit rate is very much smaller for short messages than for long messages. Third, although the messages per second increase for shorter messages, the message rate does not increase as quickly as the bit rate decreases. All of these facts indicate that performance is being constrained primarily by message copying and header manipulations within the nodal computer, not by the physical communication link. This is not surprising since the NRTS message passing systems were designed to support tracking experiments in the test bed and were not optimized for throughput. The message and bit rates provided by NRTS are sufficient for the DSN test-bed experiments. For example, tracking experiments have been executed with up to eight nodes communicating via Ethernet.

The broadcast protocol implementation was tested and debugged in two phases. First it was checked out in a laboratory test setup with two nodal computers connected back-to-back through RUI boards. The output from each RUI board was connected directly to the input of the other RUI board. The nodal computers in this configuration contained multiple processors and a full set of input-output devices. The second test situation was with single processor nodal configurations and the back-to-back connection replaced with radios.

Two major new problems were discovered and solved in the second phase, even after the code was completely operational in the back-to-back configuration. First, the monitor PROM used to load NRTS did not correctly handle bus errors that were generated when the monitor tried to initialize boards that were not present in the node. This was not a problem in the initial test setup since it was completely populated, but it was necessary to correct the problem to allow operation with less populated nodes such as those being used by the CNT radio developers. Monitor source code was modified so that, by the use of conditional assembly techniques, one copy of the monitor code now produces a version for the general DSN nodes or for the nodes used by the CNT radio developers. New monitor PROMs were made and installed in the radio nodes.

The second problem was that NRTS failed to respond quickly enough to CNT radio interrupts. This was solved by means of a combination of hardware and software changes. The radio hardware was modified to allow more time for the processing of interrupts and the interrupt routines were rewritten to achieve greater speed. In addition, two message-handling routines were rewritten to eliminate excessive message copying.

V. SYSTEM UPGRADES

DSN test-bed elements, especially those items scheduled to be deployed and used in the field, were reviewed with emphasis placed upon maximizing system reliability. A number of areas of reliability risk were identified in the course of field tests at Hanscom Air Force Base. The problems have been corrected by modifying or upgrading key components. A subsequent field test carried out on 13 to 15 January, when it happened that the equipment cycled through temperature extremes as low as +5°F, demonstrated stable, reliable system performance. Details of specific improvements are provided below.

All six data-collection nodes (three mobile units and three laboratory facilities associated with fixed microphone arrays) have been provided with new preamplifier/filter front-end arrays. These units, which condition the raw analog microphone signals prior to A/D conversion, include an amplifier with selectable gain to match signal dynamic range; a low-cut filter; a notch filter for removing 60-Hz interference; an anti-aliasing low-pass filter; and an output buffer/multiplexer. The new units, which are manufactured by Geosource, Inc., provide substantial improvements in flexibility and stability compared to the old equipment they replaced.

A new method has been worked out for applying watertight housings over the microphones used in the test bed. Occasional moisture penetration into the microphone interiors had been a problem area in the past. The new system has been tested by water immersion and by extended exposure outdoors, and has performed successfully.

In all past field measurements, a separate multiconductor cable hundreds of feet long was deployed for each of the nine microphones in a microphone array. Many problems resulted, including installation difficulty and connector unreliability. New custom designed Geosource, Inc. sensor cables have now been procured, combining all the necessary conductors in a single heavily jacketed cable with a separate (and very sturdy) breakout and connector for each microphone. One of the cables was tested during the 13 to 15 January field trials, and many advantages were apparent. Setup time was reduced to 15 to 30 min, compared to several hours with the old individual cables. Improved data quality and reliability were also observed.

The digital tape recorders in the three mobile units have been perennial trouble sources, because they were designed for a laboratory environment rather than for field service. One of them has been replaced with a new Kennedy Model 9400 ruggedized digital recorder that is better suited for field work. The new unit has performed perfectly in the January field tests and under daily use since then. Additional systems are being purchased for the other two mobile units.

Reliability problems have been experienced in the past with the PDP-11/34 computers, apparently often associated with vibration and heat. These problems have been overcome effectively by tightening up the cases, providing extra packing and bracing for the circuit boards, and insuring proper air flow.

GLOSSARY

AVSE	Air Vehicle Survivability Evaluation
CNT	Communication Network Technology
DSN	Distributed Sensor Networks
NRTS	Nodal Run-Time System
POV	Plane-of-View
RUI	Radio Unit Interface
SATS	Semiannual Technical Summary
SGI	Silicon Graphics, Inc.
SNC	Standard Nodal Computer
SPS	Sound Processing Subsystem
UIP	User Interface Program

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ESD-TR-86-085	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Distributed Sensor Networks		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical Summary 1 October 1985 — 31 March 1986
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard T. Lacoss		8. CONTRACT OR GRANT NUMBER(s) F19628-85-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, MIT P.O. Box 73 Lexington, MA 02173-0073		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element No. 62708E Project Nos. 5D30 and 5T10 ARPA Order 3345
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE 31 March 1986
		13. NUMBER OF PAGES 46
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB, MA 01731		15. SECURITY CLASS. *(of this Report)* Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

multiple-sensor surveillance system	acoustic sensors	digital radio
distributed tracking	video sensors	distributed estimation
target surveillance and tracking	low-flying aircraft	
communication network	acoustic array processing	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report describes the work performed on the DARPA Distributed Sensor Networks Program at Lincoln Laboratory during the period 1 October 1985 through 31 March 1986.